

Sensitive Data Identification with Cornell's Spider

Joshua Beeman

Information Systems and Computing

Overview

- Sensitive data
- Cornell's Spider
- Lessons from the field: SAS
- Lessons from the field: ISC

Sensitive Data

- What is it?
 - SSN's
 - Credit Cards
 - Student records
 - Medical records
 - Financial information
 - ...and more, depending on your business

Sensitive Data

- What are the risks?
 - Individual and institutional harm (e.g., identity theft, reputational damage, etc.)
 - Legislation
 - HIPAA
 - FERPA
 - Sarbanes-Oxley
 - Beginning with California Senate Bill 1386 and now 32+ more (including Pennsylvania)

Sensitive Data

On December 22nd, 2005, Pennsylvania was the 22nd state to enact a Data Breach Disclosure Law:

"personal information" includes individuals' names coupled with unencrypted information that identifies their (1) Social Security number; (2) Driver's license number or State identification card; and (3) financial account information.

The statute mandates the form of notice in the event of any "unauthorized access and acquisition of computerized data" that materially compromises the security or confidentiality of such "personal information".

<http://www.privsecblog.com/archives/state-legislation-pennsylvania-becomes-22nd-state-to-enact-a-data-breach-disclosure-law.html>

Sensitive Data

- Finding it:
 - Auditing (before a breach)
 - Forensics (after a breach)

Sensitive Data

- Examples of Tools

- Cornell Spider

<http://www.cit.cornell.edu/security/tools>

- Purdue:

<http://www.purdue.edu/securepurdue/services/scanningTools.cfm>

- U. Texas “SENF”

<https://source.its.utexas.edu/groups/its-iso/projects/senf/>

- Other commercial and free tools...

Cornell's Spider

- Versions:
 - Spider 2.9.5 for Windows (Spider3 in Beta)*
 - Spider 4.0 for Linux engine
 - Spider Beta Test for Mac OS X
 - Spider Engine for UNIX
 - Spider Simple

** Not suitable for forensics; modifies access timestamps on all files it scans.*

Cornell's Spider

- **Features**

- **Spider for Windows**

- scan regular files for any number of different confidential data types
 - assign Luhn or SSN validators on a per-regex basis
 - assign human-readable names for regular expressions
 - scan NTFS alternate data streams for any number of different confidential data types
 - scan files inside ZIP archives
 - read Excel data directly using OLE
 - create encrypted log files
 - create CSV logs for easy import to Excel or Access
 - remote logging to Event logger, UNIX syslog
 - variable lists of file extensions to scan or skip
 - unattended operation for use from task scheduler

Cornell's Spider

- **Features**
 - **Spider for Linux**
 - examine Windows or UNIX systems for files containing any number of confidential data types
 - centralized logging
 - scan files inside ZIP, RAR, ARJ, ZOO archives
 - HTML log format for easy examination of scan results

Cornell's Spider

Brief Demonstration

Cornell's Spider

- Eliminating False Positives:
 - File type exclusions (binary files such as .exe, .dll, .iso, etc.)
 - Refining Regular Expressions:
 - PA SSN: `(159|1[6-9][0-9]|20[0-9]|21[0-1])\d{6}`
 - Canadian SIN: `\d{3} \d{3} \d{3}`
 - Exclude “Fake” SSN: `[0-7]\d{2}[-\s]\d{2}[-\s]\d{4}`

State specific SSN information is here: <http://www.ssa.gov/employer/stateweb.htm>

Cornell's Spider

- Other problems reported:
 - Higher false positives in code in HTML
 - Representations of sensitive data (e.g. - pictures) are not found
 - Hybrid text/binary files on unix systems (such as MS Office files) have higher false positives

Lessons from the Field

- Examples & Lessons Learned:

Warren Petrofsky, SAS

Edwin Read, ISC

Spider on Linux/Solaris

Warren Petrofsky

School of Arts and Sciences

Spider on Linux/Solaris Basics

- Client / Server based
- Forensically sound – Does not change MAC times of scanned files like Windows version
- Can read text, zip, word, and Outlook pst files, among others
- Can be run from bootable Linux (Helix) cd (<http://helix.cit.cornell.edu/>)

Technical Tweaks

- Excluded PostScript files
- Ran Spider within a for-loop on each home directory to generate individual reports and to avoid symlink hanging
- Post-processed results to categorize accessibility of each file and give a risk ranking
- NFS mounted Solaris file systems on Linux system to make spidering possible

Lessons from a Linux Dept. Server

- Server: A departmental web, email, file, and shell server, in use for more than 10 years
- Users: 325+ faculty, staff, and graduate students
- What we found: old classlists, grades, personal financial documents

What We Did with the Results

- Triage:
 - Data available from the web, and has been indexed by major search engines (Note: We have found no instances of this yet.)
 - Data available from the web, no indexing
 - Data available to all users with accounts on the server
 - Data available to a subset of users on the server
 - Data properly restricted to user alone

What We Did...Continued

- Worked closely with faculty computing liaison and LSP
- Created per-user reports and put single text file in home directory of each user
- Asked faculty computing liaison to send out pre-announcement
- Sent targeted email to each user whose files were flagged
- Deleted report after seven days

What We Learned

- Almost all confidential university data was more than 7 years old
- Users with some confidential data are likely to have a lot
- Users with confidential data on the dept. server are likely to have confidential data on their desktops
- Few users have any idea what kind of data they have

Our Next Steps

- Provide a more intuitive way for users to view, delete, & mark files as false-positives
- Run spider on desktops of users who have confidential data on central servers
- Continue to run Spider as part of our SPIA inventory processes
- Continue to investigate additional tools

Spider on Windows

Edwin Read

Information Systems and Computing

Spider on Windows

Where we are:

- Environment: single Win2k file/print server. All project and user personal directories on one server
- Status: Still in “fact finding” stage; have run tool to determine best settings and estimate risk

Our issues

- Credit card numbers not an issue for us
- Spider only looks for formatted ssn's with delimiters (eg: 999-99-9999)
- Most of our issues are with unformatted ssn's in data files

Finding unformatted ssn's

- Created regex expression to look for unformatted ssn's in data extracts:
 - `[\s]{1}[0-9]{9}[\s]{1}|^[0-9]{9}[\s]{1}`
 - This finds 999999999 pattern either at the beginning of a line or surrounded by delimiters like tab or comma.

Issues

- Since we do not have issues with credit card numbers (proven with initial scan), eliminated them from the scan
- Ben 26 digit generated false positives (credit card)
- Fine tuning spider options speeds the scan and makes the results more usable.

Fine tuning



Results example

| PATH | CREATE TIME | MODIFY TIME | REGEX | SCORE |
|---------------|----------------------|----------------------|---------|-------|
| E:\stuff1.doc | Dec 5 2005 12:22:57 | Dec 5 2005 12:24:33 | SSN | 1 |
| e:\stuff2.doc | Jan 9 2006 12:22:52 | Jan 9 2006 14:03:27 | SSN | 1 |
| e:stuff3.doc | Feb 16 2006 17:47:37 | Feb 17 2006 17:01:43 | Raw SSN | 100 |

Conclusion

Questions?

Additional information is also available by emailing
security@isc.upenn.edu or privacy@isc.upenn.edu.